IN THE CLAIMS:

The status of the claims is as follows:

1. (Currently Amended) A method of managing processor instructions, the method comprising:

decoding a first instruction into a plurality of operations with a decoder;

passing a first copy of the operations from the decoder to a build engine associated with a trace cache;

passing a second copy of the operations from the decoder directly to a back end allocation module, the operations bypassing the build engine and the allocation module being in a decoder reading state; and

determining at the decoder whether a trace cache reading resume condition is present based on a second instruction, the determining being conducted before the second instruction leaves the decoder.

2. (Previously presented) The method of claim 1 further including:

switching the allocation module from the decoder reading state to a trace cache reading state if the resume condition is present.

3. (Original) The method of claim 2 further including searching the trace cache for an instruction pointer that corresponds to the second instruction.

4. (Original) The method of claim 3 further including repeating the searching for every subsequent instruction decoded by the decoder.

5. (Original) The method of claim 3 further including repeating the searching for a subset of every subsequent instruction decoded by the decoder.

6. (Original) The method of claim 5 further including selecting the subset based on a lookup to an abbreviated tag array.

7. (Original) The method of claim 5 further including selecting the subset based on an instruction heuristic.

8. (Original) The method of claim 7 further including determining whether an immediately preceding instruction was a branch instruction.

9. (Original) The method of claim 1 further including passing the second copy to a buffer of the allocation module.

10. (Withdrawn) A method of managing processor instructions, the method comprising:

determining at a decoder whether a resume condition is present based on an instruction; and

switching a backend allocation module of a processor from a decoder reading state to a trace cache reading state when the resume condition is present.

11. (Withdrawn) The method of claim 10 further including searching the trace cache for an instruction pointer that corresponds to the instruction.

12. (Withdrawn) The method of claim 11 further including repeating the searching for every subsequent instruction decoded by the decoder.

13. (Withdrawn) The method of claim 11 further including repeating the searching for a subset of every subsequent instruction decoded by the decoder.

14. (Withdrawn) The method of claim 13 further including selecting the subset based on a lookup to an abbreviated tag array.

15. (Withdrawn) The method of claim 13 further including selecting the subset based on an instruction heuristic.

16. (Currently amended) A method of managing instructions in a computer processing architecture having a trace cache, the method comprising:

detecting a trace cache miss;

receiving a first instruction;

decoding the first instruction into a plurality of operations with a decoder;

passing a first copy of the operations from the decoder to a build engine associated with the trace cache;

passing a second copy of the operations from the decoder directly to a back end

allocation module, the operations bypassing the build engine and the allocation module being in a decoder reading state;

receiving a second instruction;

determining at the decoder whether a resume condition is present based on the second instruction, the determining being conducted before the second instruction leaves the decoder; and

switching the allocation module from the decoder reading state to a trace cache reading state if the resume condition is present.

17. (Original) The method of claim 16 further including searching the trace cache for an instruction pointer that corresponds to the second instruction.

18. (Original) The method of claim 17 further including repeating the searching for every subsequent instruction decoded by the decoder.

19. (Original) The method of claim 17 further including repeating the searching for a subset of every subsequent instruction decoded by the decoder.

20. (Currently Amended) A processor instruction management system comprising:

a decoder to decode a first instruction into a plurality of operations; and

a controller to pass a first copy of the operations from the decoder to a build engine associated with a trace cache, the controller to pass a second copy of the operations from the decoder directly to an allocation module, the operations bypassing the build engine and the

allocation module being in a decoder reading state, the controller including control logic to determine [at the decoder] whether a trace cache reading resume condition is present based on a second instruction before the second instruction leaves the decoder.

21. (Previously presented) The instruction management system of claim 20 wherein the control logic is to switch the allocation module from the decoder reading state to a trace cache reading state if the resume condition is present.

22. (Original) The instruction management system of claim 21 wherein the control logic is to search the trace cache for an instruction pointer that corresponds to the second instruction.

23. (Original) The instruction management system of claim 22 wherein the control logic is to repeat the searching for every instruction decoded by the decoder.

24. (Original) The instruction management system of claim 22 wherein the control logic is to repeat the searching for a subset of every instruction decoded by the decoder.

25. (Original) The instruction management system of claim 24 wherein the controller further includes an abbreviate tag array, the control logic selecting the subset based on a lookup to the abbreviated tag array.

26. (Original) The instruction management system of claim 24 wherein the controller further includes an instruction heuristic, the control logic selecting the subset based on the instruction heuristic.

27. (Original) The instruction management system of claim 26 wherein the instruction heuristic determines whether an immediately preceding instruction was a branch instruction.

28. (Original) The instruction management system of claim 20 wherein the decoder and the controller are part of an application specific integrated circuit (ASIC).

29. (Canceled).

30. (Canceled).

31. (Canceled).